3M-8

GAを用いた分散システム向けのEnd-to-Endタスクの割り当て

枝光　圭祐†　　兪　明連†　　横山　孝典†
東京都市大学†　東京都市大学†　東京都市大学†

1　はじめに

現在，自動車などの制御は組込みシステムによって行われており，多くは分散システムの形態をとっている．分散システムで実行されるタスクは，一つのノード上のタスクだけで処理が完了して機能するものだけではなく，他のノード上のタスクの実行結果を参照して動作し機能するタスクも存在する．分散システムでは一つの機能を満たすタスクの集合をEtE(End-to-End)タスクと呼び，EtEタスクを構成する要素となるタスクをサブタスクと呼ぶ．本研究では分散システムのタスクの割り当てについて，二つ以上のサブタスクで構成されたEtEタスクの中で最大の応答時間を持つEtEタスクの応答時間を小さくする割り当て解を最適化問題の手法の一つであるGAを用いて求める．

2　システムモデル

本研究ではサブタスク一つだけで構成されるEtEタスクと二つ以上のサブタスクで構成されるEtEタスクが混在したタスクセットの割り当てを対象とする．さらに二つ以上のサブタスクで構成されるEtEタスクのサブタスクの数は2個から最大で5個とし，同じEtEタスクを構成するサブタスクのうち連続するものは同じノードには割り当てないものとする．

3　関連研究

3.1　スケジューリング

分散システムでのタスクの実行スケジューリングは周期の短いタスクに高い優先度を与えるRMで行われる．ただし，実行されるタスクはEtEタスクでありEtEタスクのサブタスクのリリースにおいて，そのサブタスクが参照するサブタスクの処理が完了した後にリリースされるようにしなければならない．これに従来研究としてRGプロトコルが提案されている．

RG(Release Guard) プロトコル [1]

RGプロトコルとは，分散システムのノード間での通信プロトコルとして提案された手法で，他のサブタスクを参照するサブタスクのリリースをRGと呼ばれるものを用意して制御する．RGとは制御するサブタスクが参照するサブタスクの実行が完了するよりも前にリリースされてしまわないようにするためのもので，以下のように設定される．

(a) 初期値は0とする．

(b) サブタスクがリリースされたとき，RGをリリースされた時刻とサブタスクの周期を足した時刻に更新する．

(c) サブタスクがあるプロセッサがアイドル状態ならば，リリース時刻をその時刻に更新する．

図1にRGプロトコルを使用したスケジューリング例を示す．実行するタスクセットとプロセッサ数は　T={$T_{1.1}$=(2,4)，$T_{2.1}$=(2,6)，$T_{2.2}$=(2,6)，$T_{3.1}$=(15,24)}，M=2とし，$T_{1.1}$と$T_{2.1}$はノード1(P1)，$T_{2.2}$と$T_{3.1}$はノード2(P2)に割り当てられている．また，$T_{1.1}$と$T_{3.1}$はサブタスクが一つのEtEタスクであり，$T_{2.1}$と$T_{2.2}$の二つで一つのEtEタスクを構成し，$T_{2.2}$は$T_{2.1}$の結果を参照するサブタスクである．

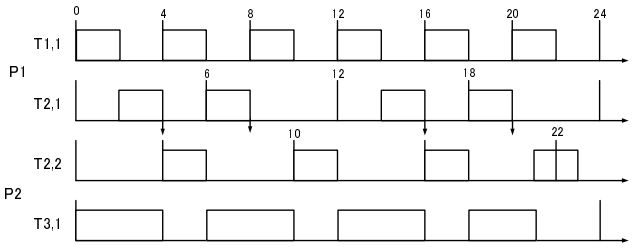

図1: RGプロトコルシミュレーション例

図1をみると$T_{2.2}$は，RGの(a)より初期値が0なので$T_{2.1}$の処理が完了してすぐにリリースされている．

その後は(b)に従って$T_{2.2}$のリリースが行われている．そして時刻21においてP2がアイドル状態になるので(c)が適用されて22だったリリース時刻が21に更新されている．

3.2　割り当て

従来研究ではタスクの割り当ては任意で行われていて決まった手法は提案されていない．

4　GA(遺伝的アルゴリズム) による解法

タスクのノードへの割り当て解を求める問題は組合せ問題である．組合せ問題は解の候補があればあるほど時間がかかる問題とされている．またタスクの割り当て解を求める問題はNP困難であるため実用的な時間で解くことが非常に難しい．そのため実用的な時間で近似解を求めることのできるGAを用いることとする．GAとは問題に対する解の候補を遺伝子の並びで

ある染色体に見立てて，計算機上で生物進化の過程を模倣した最適化の手法である．染色体が解に対応して，染色体の集合を交叉，突然変異，選択させることで近似解を求める．GA の流れを図 2 に示す．最初に初期集団の生成から始まり，交叉，突然変異，選択の操作を終了条件を満たすまで繰り返す．

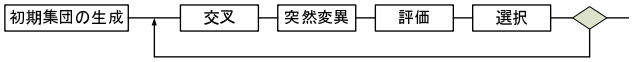

図 2: GA の流れ

4.1 初期集団の生成

初期集団とはランダムに生成された染色体の集まりである．問題の解の候補を遺伝子の並びに変換して染色体とし，指定した数だけ用意して初期集団とする．

染色体の構造は 2 つ以上のサブタスクで構成される EtE タスクを前に並べ，サブタスク一つのみで構成される EtE タスクを後ろに並べて遺伝子列をつくる．図 3 は本研究での染色体構造を示したものである．

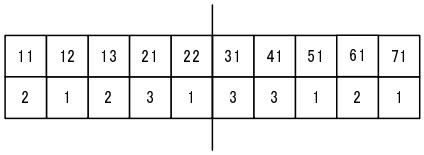

図 3: 染色体構造

4.2 適応度

適応度とは染色体が環境にどれだけ適応しているか，つまり求める最適解に近づいているかを表す数値である．本研究では二つ以上のサブタスクで構成される EtE タスクの最大の応答時間を $\boldsymbol{f(x)}$ として $\boldsymbol{f(x)}$ が小さいほど適応度が高くなるように適応度 $\boldsymbol{g(x)}$ を以下の式で表すものとする．

$$\boldsymbol{g(x)} = \frac{1}{\boldsymbol{f(x)}} \tag{1}$$

4.3 交叉

交叉とはランダムに選ばれた 2 個の親染色体を交叉確率 (交叉が起こる確率) により遺伝子列を入れ替え子染色体を生成する操作である．本研究では染色体の遺伝子列のある一点を選びそれよりも後ろにある部分を入れ替える一点交叉で交叉を行い，入れ替えをする点は二つ以上のサブタスクで構成される EtE タスクが並ぶ部分と一つだけのサブタスクで構成される EtE タスクが並ぶ部分の間とする．図 4 は本研究の交叉の例を示したものである．左の二つが親染色体から本研究で設定した点を境に列を入れ替えることで右の二つの子染色体を生成している．

4.4 突然変異

突然変異とは突然変異確率 (突然変異が起こる確率) により染色体の遺伝子列の一部を変化させて新しい染色体を生成する操作である． この操作は交叉のみでは局所解に収束してしまう可能性があるため，それを回避するためのものである． 本研究では EtE タスクの最初のサブタスクの中からランダムに一つ選びその部分を変化させる．突然変異は遺伝子列のすべての EtE タスクの最初のサブタスクを対象とする．

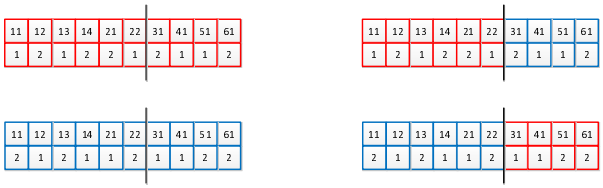

図 4: 交叉の実行例

4.5 選択

選択とは集合の各染色体の適応度にもとづいて次世代に残す染色体を選ぶ操作である．適応度が高いほど子孫を残しやすく小さいものほど死滅しやすい．本研究では GA の選択の方法として一般的に使われるルーレット選択を用いた．ルーレット選択とは適応度に比例した割合で選択する方法である．この操作を集団数が得られるまで繰り返す．

5 評価

ランダムにシステム利用率 10%から 100%まで，さらに各利用率の区分ごとにタスクセットの中で複数のサブタスクで構成される EtE タスクの割合が 10%から 100%までそれぞれ 1000 セットずつ用意し，GA によって解を求める．求めた解を RG プロトコルを用いてシミュレーションしランダムに割り当てを行った場合と比較をして評価する．比較する項目は複数のサブタスクで構成される EtE タスクの最大の応答時間である．シミュレーションを行った結果，ランダムに割り当てた場合に比べて複数のサブタスクで構成される EtE タスクの最大の応答時間を小さくする割り当て解を求めることができた．

6 まとめ

本研究では分散システムでの複数のサブタスクで構成される EtE タスクを含むタスクセットの割り当てについて，複数のサブタスクで構成される EtE タスクの最大の応答時間を小さくする割り当て解を GA を用いて求めた．

参考文献

[1] JUN SUN, “ FIXED-PRIORITY END-TO-END SCHEDULING IN DISTRIBUTED REAL-TIME SYSTEMS ”,Thesis of Doctor of the University of Illinois at Urbana-Champaign (1997).